

Obtaining the Sonnet Example Files

This document contains a list of all the example files which came with your Sonnet software. Each entry contains an example name, a list of file(s), a picture of the example circuit and a brief description.

For single file examples, it is possible to open the example in the project editor, then perform a Save As to copy the example to your working directory.

For multiple file examples, there is a copy command which copies the set of example files to your working directory.

To obtain an example, follow the procedure below.

NOTE:

You must have Adobe Acrobat Reader installed on your system to access the manuals in PDF format. If you do not have the program, it is available for installation by selecting *Admin* ⇒ *Install Acrobat* from the Sonnet Task Bar main menu.

1 Click on the Manuals button on the Sonnet Task Bar.

The “Sonnet User’s Manuals” is opened on your display. The manuals available in PDF format are identical to the hard copy manuals which came with your installation package.

If you do not know how to invoke the Sonnet task bar, please refer to ["Invoking Sonnet"](#) in the **Getting Started** manual.

2 Click on the Application Examples button in the PDF document.

This opens the Sonnet Application Examples page. You may also select *Help* ⇒ *Examples* from any program menu to reach this page.

3 Click on the “Complete List” button.

A page appears with a complete list of all available files in alphabetical order.

4 Click on the name of the example you want.

This will take you to the example project in the Application Examples manual. If your example has a single file, continue at step 5. If you are obtaining a multi-file example, continue at step 7.

5 For a single file example, click on the Load into Project Editor button at the top of the page.

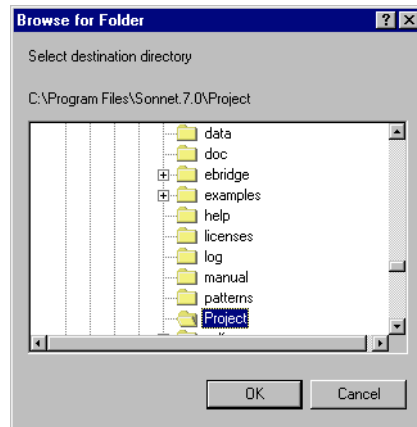
The project editor is invoked on your display with your chosen example file open.

6 Select *File* ⇒ *Save As* from the project editor menu to save the file to a working directory.

The example files are read only. You must use the browse window which appears as a result of the Save As command, to save the file to another location. A copy of an example file should always be used to avoid corrupting the example sources. This completes your obtaining the example

7 For a multiple file example, click on the Copy Example button at the top of the page.

Adobe Acrobat will prompt you as to whether you wish to launch this application. Click on Yes. A command window appears briefly on your display, followed by a browse window.

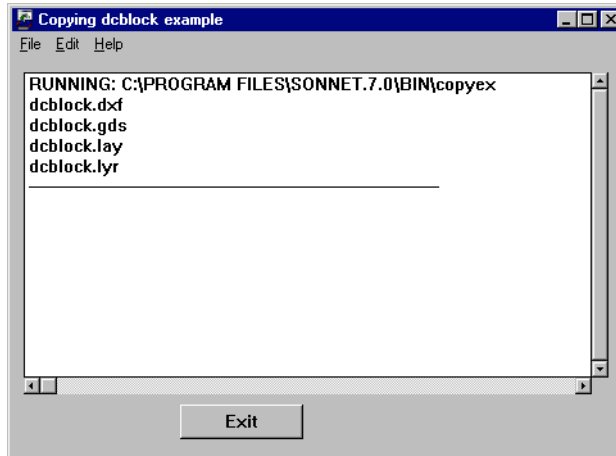


8 Use the Browse window to select your working directory.

Your working directory is where you keep your Sonnet project files.

9 Click on the OK button to copy the files to your working directory.

The example directory which contains the example files is copied to the specified directory. An output window appears to confirm the copy as shown below for the example Dcblock.

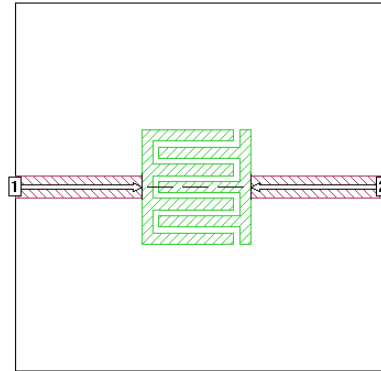


10 Once the files are copied, click on the Exit button to close the output window.

Interdigitated Capacitor



NAME: Idig_cap



LICENSE REQ'D: Sonnet Lite

CATEGORY: Resistors, Capacitors and Inductors

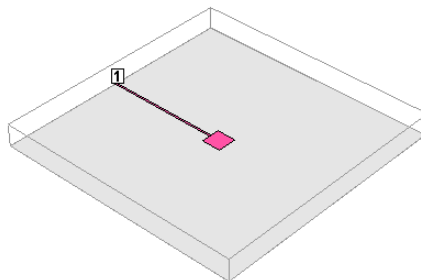
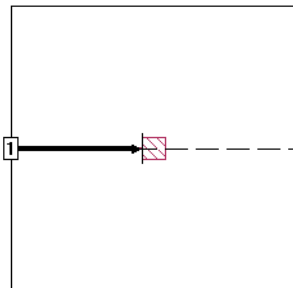
FILE(S): idig_cap.son

DESCRIPTION: This example is an interdigitated capacitor composed of seven closely coupled parallel lines. This capacitor has a nominal value of about 0.1 pF below 1 GHz. As the frequency increases, the capacitor becomes more capacitive, due to the increase in fringing fields between the fingers. This type of behavior is typical of microstrip interdigitated capacitors.

Simple Patch Antenna



NAME: Uspatch



LICENSE REQ'D: Sonnet Lite

CATEGORY: Antennas

FILE(S): uspatch.son

DESCRIPTION: This is a very simple patch antenna example using 25 mil Rogers RT5880. The patch antenna is fed with a microstrip feed, but a via feed could also be used.

The patch is 4340 by 4340 mils. Rather than using a cell size of 10 by 10 mils, a cell size of 217 by 217 mils was chosen. This value goes into 4340 evenly, allowing the patch dimensions to be exact but using less memory (and running faster) than the same circuit with a 10 by 10 mil cell size. Remember, smaller cell size gives a more accurate simulation but requires more memory. Also, the cell size does not have to be an integer number (86.8 would be fine) and the X cell size does not need to be equal to the Y cell size.

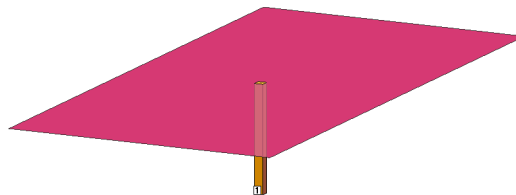
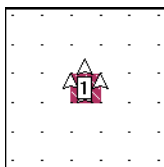
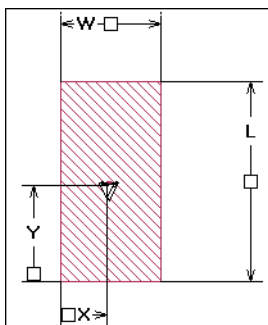
You can also use the Cell Size calculator which automates the process of selecting a cell size. This can be found by selecting *Circuit* \Rightarrow *Box* and clicking on the Cell Size Calculator button. Further information on this feature can be obtained by clicking on the Help button.

NOTE: Before analyzing an antenna, you should read the [Chapter 20, "Far Field Viewer Tutorial"](#) in the Sonnet User's Guide. This gives you critical information concerning modeling radiation with Sonnet.

Patch Antenna with Via Feed



NAME: Feedvia_patch



LICENSE REQ'D: Sonnet Lite

CATEGORY: Antennas

FILE(S): feedvia_patch.son

DESCRIPTION: Many patch antennas are fed with a coax feed. The outer conductor of the coax is connected to the ground plane, and a small hole is drilled to allow the center conductor to pass through the ground and continue through the substrate to be connected to the bottom of the patch metalization

You can approximate this configuration in Sonnet by using a via-port. The Sonnet analysis of a via-port includes the inductance of the via, which should be similar to the inductance of the coax. This project shows a simple patch antenna which uses a via-port to feed the antenna.

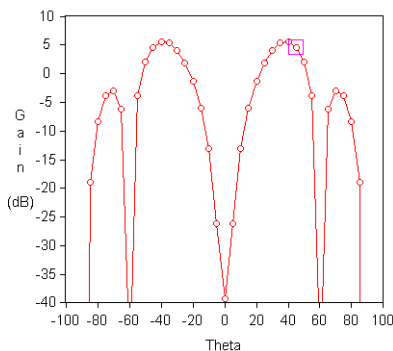
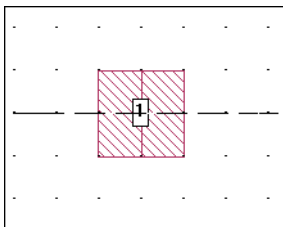
In this project file you will see that we parameterized both the patch size and X,Y location of the via feed point. The analysis is set to sweep the X position from the edge of the patch to the middle. It could easily be changed to sweep any combination of these four parameters.

NOTE: Before analyzing an antenna, you should read the “Antennas and Radiation” chapter in the Sonnet User’s Manual. This gives you critical information concerning modeling radiation with Sonnet.

Infinitesimal Dipole Above a Ground Plane



NAME: Infpole



LICENSE REQ'D: Sonnet Lite

CATEGORY: Antennas

FILE(S): infpole.son

DESCRIPTION: This is an example of an infinitesimal dipole one wavelength above a ground plane at 1 GHz. This means there is an image one wavelength below the ground plane. The results may be compared to the exact theoretical antenna pattern as shown in [this figure](#) in the **Sonnet User's Guide**.

Although this example is not very practical, it is a good example to use for validation because of its simplicity. Infinitesimal dipoles are also used to represent capacitor-plate, or top-hat-loaded antennas, as well as being used as building blocks for more complex circuits.

[Continue](#)

Infinitesimal Dipole Above a Ground Plane (Cont'd)



The geometry was created by drawing two small polygons next to each other and placing an internal port between them. The internal port can be thought of as providing a negative voltage to the left polygon, and a positive voltage to the right polygon. Thus the port provides a small source of horizontal current. The dipole is placed in a very large box so that the box walls do not interfere with the radiation of the dipole. See [Chapter 19, "Antennas and Radiation"](#) in the **Sonnet User's Guide** for rules on setting up antenna problems.

This circuit is used in [Chapter 20, "Far Field Viewer Tutorial"](#) in the **Sonnet User's Guide**.

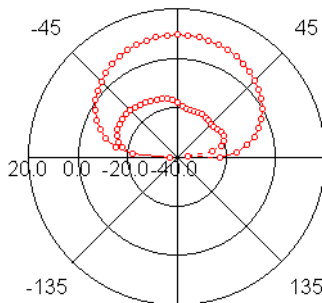
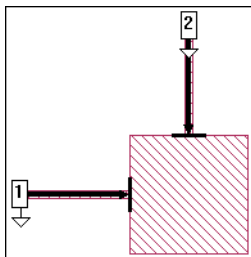
[Back](#)

[Next Example](#)

Circularly Polarized Patch Antenna



NAME: Rhp



LICENSE REQ'D: Sonnet Lite

CATEGORY: Antennas

FILE(S): rhp.son

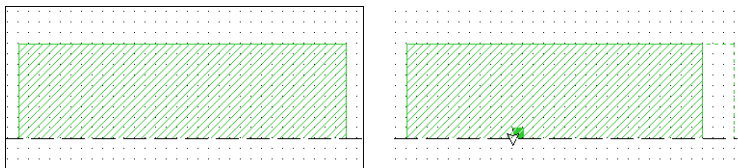
DESCRIPTION: This example is a patch antenna with two ports, which when excited 90 degrees out of phase, produce a circularly polarized antenna pattern.

TIP: To generate the circularly polarized antenna pattern, open rhp.son in the far field viewer and select a polar plot. Select *Graph* \Rightarrow *Calculate* \Rightarrow *Ports* from the main menu and excite port 1 with 1 volt and port 2 with 1 volt at 90 degrees. Then select *Graph* \Rightarrow *Polarization* \Rightarrow *Circular* and check the E RHP and E LHP checkboxes.

Dual Patch Microstrip Antenna



NAME: Dual_patch



LICENSE REQ'D: Sonnet Lite

CATEGORY: Antennas

FILE(S): dual_patch.son

DESCRIPTION: This example is a two-layer microstrip patch antenna, taken from APS Trans, May 1990, pp. 613-614, Fig. 8a. The lower patch is resonant at about 3.1 GHz, and the upper patch resonates at about 3.5 GHz.

The lower patch is fed with a via port, a good approximation to a coax feed from the bottom of the antenna. Since this is a via-port, the port is not de-embedded. Thus the inductance of the port is included in the analysis results.

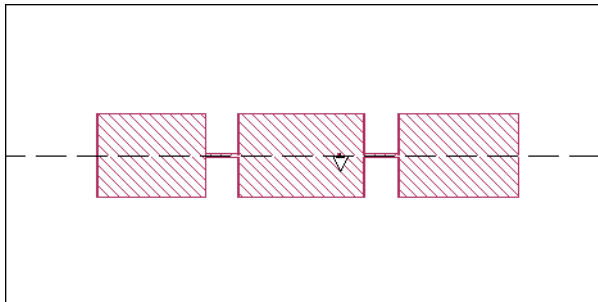
This circuit uses symmetry, so only the top half of the circuit is captured. This means that the square via port will be reflected below the line of symmetry, creating a rectangular source (aspect ratio of 2:1).

TIPS: As with any antenna problem, a large box is important for accurate results. Using a binary box can speed up the analysis of large boxes.



Triple Patch Antenna

NAME: Tripat



LICENSE REQ'D: Sonnet LitePlus

CATEGORY: Antennas

FILE(S): tripat.son

DESCRIPTION: This example shows a triple patch element. Each patch has a slightly different resonant frequency, 2.5 GHz for the left patch, 2.1 GHz for the central patch, and 2.3 GHz for the right patch. The multiple resonances results in an increased bandwidth. The antenna is fed from below with a coax probe attached to the central patch. This is accomplished in Sonnet with a via-port.

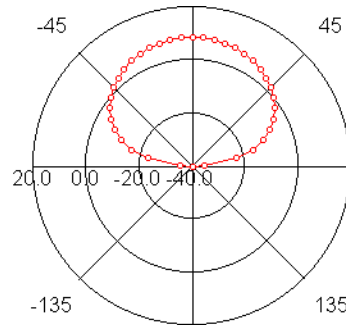
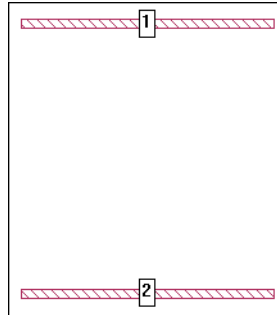
The antenna geometry was provided to Sonnet courtesy of E. Ongareau, Matra Defence. The geometry represents the design before using Sonnet to improve it.

TIPS: As with any antenna problem, a large box is important for accurate results. Using a binary box can speed up the analysis of large boxes.

For a more detailed description of this antenna, see "[Validation Example](#)" in Chapter 19, "Antennas and Radiation" in the **Sonnet User's Guide**.

Two Dipole Array

NAME: Twoarray



LICENSE REQ'D: Sonnet Lite

CATEGORY: Antennas

FILE(S): twoarray.son

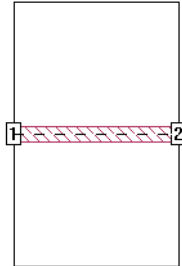
DESCRIPTION: This example is composed of two half-wave dipoles spaced approximately one-half wavelength apart. The dipoles are placed 1/10 of a wavelength above a ground plane. Because of the spacing between the dipoles a null will appear in the antenna pattern. In the post-processor viewing program, the far field viewer, the port 1 dipole is excited 53° in advance of the port 2 dipole for a total phase difference of 180° at $\theta = 45^\circ$ which provides a null at approximately $\theta = 45^\circ$ degrees. This is done by entering a source phase of 53 degrees for port 2 in the Calculations dialog box in the far field viewer (*Graph* \Rightarrow *Calculate*, then click on the Ports tab).

TIP: Since the radiation pattern for dipoles can be toward the sidewalls of the Sonnet box, the results may not be as accurate as other antennas. A large box helps decrease the error.

100 Ohm Stripline Benchmark



NAME: S100



LICENSE REQ'D: Sonnet Lite

CATEGORY: Accuracy Benchmark

FILE(S): s100.son

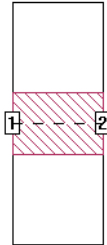
DESCRIPTION: This is an example of a lossless 100 ohm stripline thru. This circuit is used for benchmarking simulator accuracy. The dimensions of this circuit are such that the characteristic impedance is exactly 100 ohms, and the phase length is exactly -90 degrees. Thus, any deviation in the simulator results from 100 ohms and -90 degrees is error.

TIPS: Notice that the cell size in this example has been chosen such that there are 16 cells across the width of the thru. As discussed in the [Chapter 23, "Accuracy Benchmarking"](#) of the **Sonnet User's Guide**, this will result in approximately 1% error in the computed characteristic impedance. Notice, however, that this circuit also has symmetry enabled. Thus, the actual analysis is performed with only 8 cells across the width of the thru. It is important to note here that enabling symmetry does not reduce accuracy. You get exactly the same answer whether you analyze the full thru with symmetry disabled or you analyze the half thru with symmetry enabled.

25 Ohm Stripline Thru



NAME: S25



LICENSE REQ'D: Sonnet Lite

CATEGORY: Accuracy Benchmark

FILE(S): s25.son

DESCRIPTION: This is an example of a lossless 25 ohm stripline thru. This circuit is used for benchmarking simulator accuracy. The dimensions of this circuit are such that the characteristic impedance is exactly 25 ohms, and the phase length is exactly -90 degrees. Thus, any deviation in the simulator results from 25 ohms and -90 degrees is error.

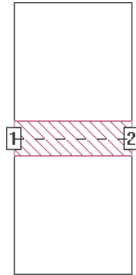
TIPS: Notice that this circuit is symmetric about the x-axis and that symmetry has been enabled. When symmetry is enabled, all metallization below the line of symmetry is ignored. Thus, it is not even necessary to draw metallization below the line of symmetry.

This circuit is used in [Chapter 23, "Accuracy Benchmarking"](#) in the **Sonnet User's Guide**.

50 Ohm Stripline Thru



NAME: S50



LICENSE REQ'D: Sonnet Lite

CATEGORY: Accuracy Benchmark

FILE(S): s50.son

DESCRIPTION: This is an example of a lossless 50 ohm stripline thru. This circuit is used for benchmarking simulator accuracy. The dimensions of this circuit are such that the characteristic impedance is exactly 50 ohms, and the phase length is exactly -90 degrees. Thus, any deviation in the simulator results from 50 ohms and -90 degrees is error.

TIPS: The top and bottom box covers in this example are lossless, grounded metal. They serve as the stripline ground metal. Thus, when analyzing stripline, it is not necessary to add extra metallization layers for the ground metal. Simply add your conductor circuitry in between and use the top and bottom box covers as the ground metal. Since the top and bottom covers do not add any subsections to the analysis, your simulation will be more efficient than if you added separate metallization layers for the ground metal.

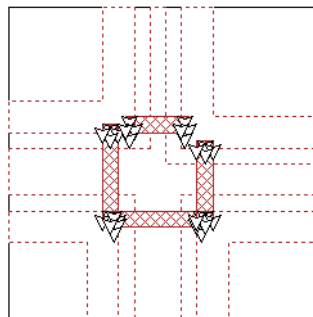
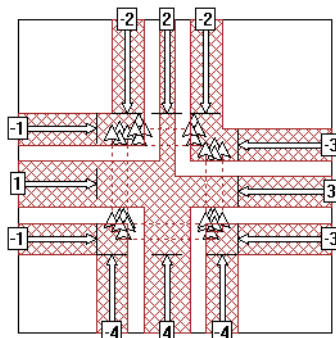
This circuit is used in [Chapter 23, "Accuracy Benchmarking"](#) in the **Sonnet User's Guide**.

Coplanar Waveguide Cross Junction



NAME:

Cocross



LICENSE REQ'D: Sonnet Professional

CATEGORY: Coplanar

FILE(S): cocross.son

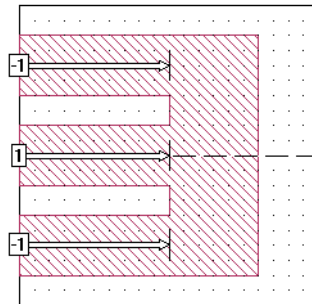
DESCRIPTION: This is an example of a coplanar waveguide cross junction. The circuit includes airbridges to suppress the slot line mode which would otherwise propagate in the coplanar transmission line.

TIPS: This is an example of an ungrounded coplanar configuration. There is no ground metal beneath the substrate. For a discussion of common coplanar configurations and how to analyze them in Sonnet, see **Chapter 22, "Package Resonances"** in the **Sonnet User's Guide**.

Coplanar Waveguide Short Circuit with Symmetry



NAME: Cosht_sy



LICENSE REQ'D: Sonnet Lite

CATEGORY: Coplanar Waveguide

FILE(s): cosht_sy.son

DESCRIPTION: This is an example of a coplanar waveguide (CPW) short circuit. CPW short circuits are often used as measurement calibration standards. Notice that both ground ports have been labeled “-1”, while the signal port is labeled “1”. This represents a balanced or “push-pull” port. Balanced ports are used because this is ungrounded CPW (the -1 port provides the ground).

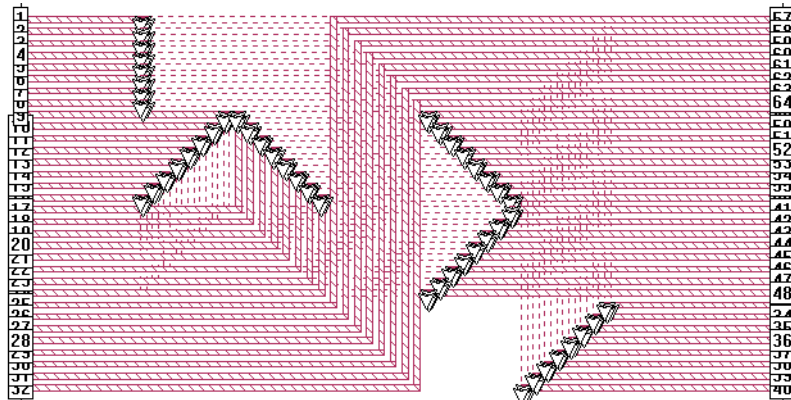
TIP: Symmetry may be used in this example even though the -1 ports are not on the line of symmetry. This is acceptable because the matching port below the line of symmetry is the same port number.

Byte Reversal Network



LOAD INTO
PROJECT EDITOR
(READ ONLY)

NAME: Br32



LICENSE REQ'D: Sonnet Professional

CATEGORY: Digital

FILE(s): br32.son

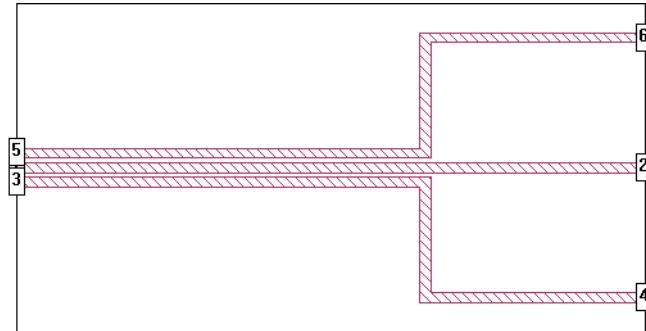
DESCRIPTION: This example is a simple byte-reversal network. There are 32 input ports and 32 output ports. The first eight bits (ports 1-8) go into the circuit, down to the second level, underneath all the other lines, and come up on the right side as the last eight bits. Thus, the byte order on input is reversed on the output.

This circuit is used as an example of obtaining a SPICE lumped element model using Sonnet. For details, see "PI Spice - A High Speed Digital Example" under **Tips and App Notes** in online help.

Coupled Printed Circuit Traces



NAME: Pspice



LICENSE REQ'D: Sonnet LitePlus

CATEGORY: Digital

FILE(S): pspice.son

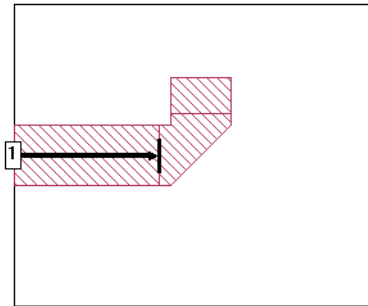
DESCRIPTION: This example contains three coupled printed circuit traces. The traces on the right would eventually be connected to three terminals of a printed circuit board mounted device. The crosstalk between the lines can be analyzed in Sonnet and a SPICE equivalent circuit can be generated which includes the coupling and discontinuities.

TIPS: Notice that symmetry was not used in the circuit. This is because symmetry cannot be used when there are ports off the line of symmetry which do not have the same number.

Open End Closely Interacting with Mitered Bend



NAME: Openmite



LICENSE REQ'D: Sonnet Level2 Basic

CATEGORY: Discontinuities

FILE(S): openmite.son

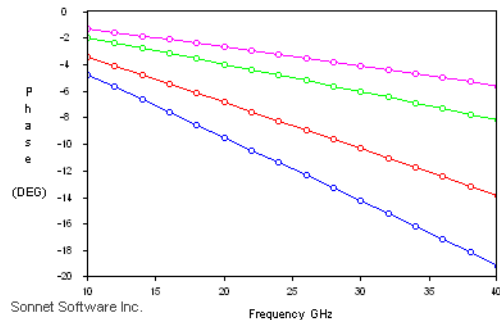
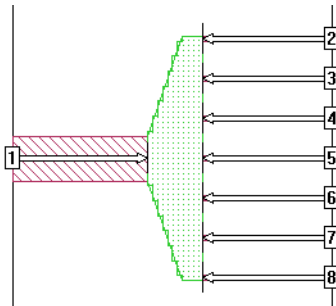
DESCRIPTION: This example contains a mitered bend in close proximity to an open end. The two discontinuities are close enough such that modeling them separately may not be appropriate. This example uses diagonal fill to represent the diagonal edge better than the standard staircase fill.

TIP: Note how the mitered region is a separate polygon. A polygon with tabs, instead of a triangle with acute angles, was used to prevent the vertices being cut off by the actual metal fill. When using diagonal fill, if the mitered region is captured as a simple triangle, without tabs, the vertices with acute angles are cut off.



FET Manifold

NAME: Fet_manifold



LICENSE REQ'D: Sonnet Level3 Gold

CATEGORY: Techniques

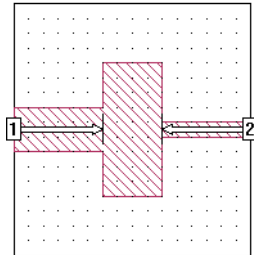
FILE(S): fet_manifold.son

DESCRIPTION: This is an example of a FET manifold. At high frequency, the phase at the outer fingers is significantly different from the phase at the center. This can be seen by comparing the slopes of S21 (bottom line) and S51 (top line). The picture above shows how the electrical length (negative phase) is related to the distance each finger is from the center finger.

TIP: Notice how the reference planes on the right are set to remove the entire length of feed line. This was done because it is assumed that the model for the FET includes these lines. Whenever there is more than one port on a box wall, the reference planes also remove all of the coupling between the lines. This is important, because if it did not, this coupling would be included twice in the final analysis (assuming a FET model which already contains the coupling is connected to this circuit).

Two Cascaded Steps

NAME: Steps



LICENSE REQ'D: Sonnet Lite

CATEGORY: Techniques

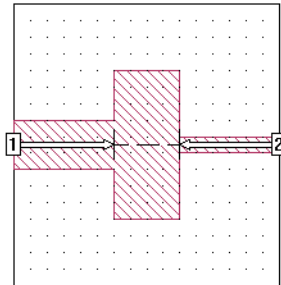
FILE(S): steps.son

DESCRIPTION: This is an example of two closely spaced step discontinuities. With circuit theory, the middle section might be modeled as two open stubs, or as a wide (low impedance) transmission line. Sonnet analyses are not based on a model, and hence can accurately model an arbitrary shape.

Two Cascaded Steps with Symmetry



NAME: Ste_sym



LICENSE REQ'D: Sonnet Lite

CATEGORY: Techniques

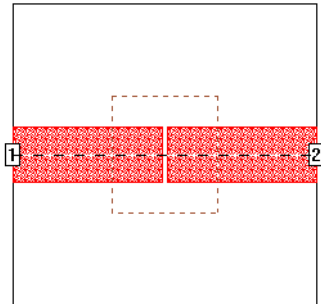
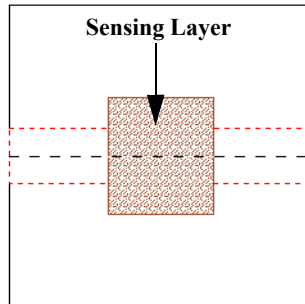
FILE(S): ste_sym.son

DESCRIPTION: This example shows the [steps](#) project with symmetry. Since the circuit is exactly the same above and below the line of symmetry (including port excitation), symmetry may be invoked to reduce memory and processing requirements. Typically, using symmetry uses one fourth the amount of memory and runs up to eight times faster with no degradation in accuracy.

Transmission Line with 20 mil Gap and Sense Layer



NAME: Tane



LICENSE REQ'D: Sonnet Lite

CATEGORY: Techniques

FILE(S): tane.son

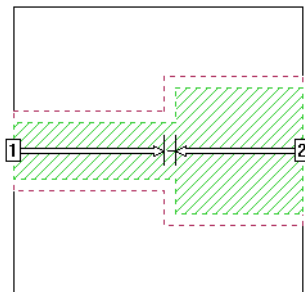
DESCRIPTION: This example illustrates how tangential electric fields may be viewed in Sonnet. The circuit contains a 180 mil wide microstrip transmission line with a 20 mil gap. A rectangular sheet of metal has been placed in the air 0.10 mils above the transmission line. This metal uses the Sense Metal metal type in the sense layer. The idea is that this metal will “sense” the tangential electric field at 0.10 mils above the transmission line, but will not influence the behavior of the transmission line.

TIP: When you define a metal type using the Sense Metal definition, Xdc is set to 0 initially; you must enter a large value for this metal so that it does not affect the rest of the circuit. A value of 1.0e6 is recommended.

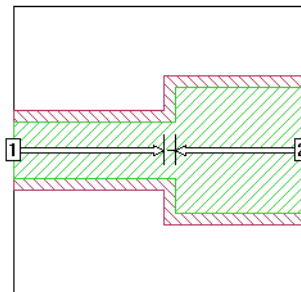
Step Junction with Thick Metal



NAME: Thkstep



Level 0



Level 1

LICENSE REQ'D: Sonnet Level2 Basic

CATEGORY: Techniques

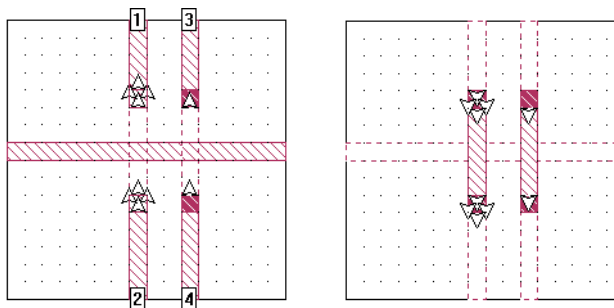
FILE(S): thkstep.son

DESCRIPTION: The example shows a step project with 3 mils of thick metal. The thick metal is placed on top of a wider zero-thickness metal to simulate a trapezoidal cross section.

Example Illustrating an Airbridge



NAME: Bridge



LICENSE REQ'D: Sonnet Lite

CATEGORY: Multilayer Circuits and Vias

FILE(S): bridge.son

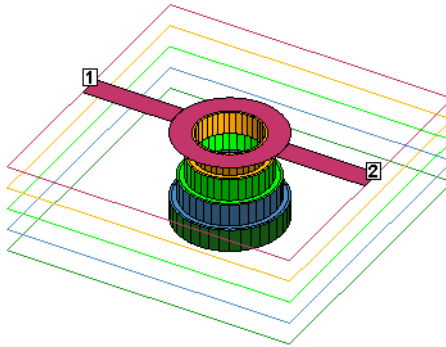
DESCRIPTION: This example demonstrates how to use vias to create a simple airbridge in a circuit where a transmission line bridges over another transmission line.

TIP: The triangle shows the direction of the via. A triangle pointing to the top of your screen means the via extends from the present level to one level above. A triangle pointing to the bottom of your screen means the via extends down one level.



Conical Via

NAME: Cvia



LICENSE REQ'D: Sonnet Professional

CATEGORY: Multilayer Circuits and Vias

FILE(S): cvia.son

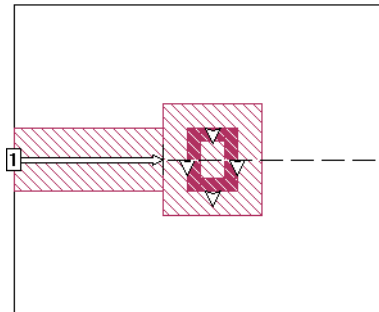
DESCRIPTION: This example is a conical via connected to ground in shunt with a microstrip line. Multiple levels were used to approximate the conical shape.

TIP: If you are modeling a large circuit with multiple vias, you may need to use a simpler model for efficiency reasons (for example, [via.son](#)). Or, you may want to create ports in place of the vias, and use a network file to connect an accurate via model to the circuit.

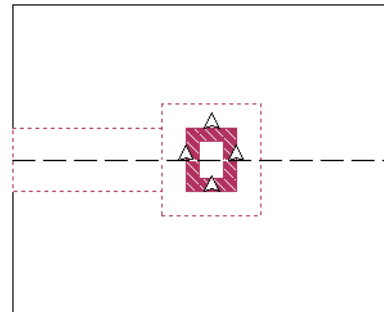
Example Illustrating a Simple Via to Ground



NAME: Via



Level 0



Ground

LICENSE REQ'D: Sonnet Lite

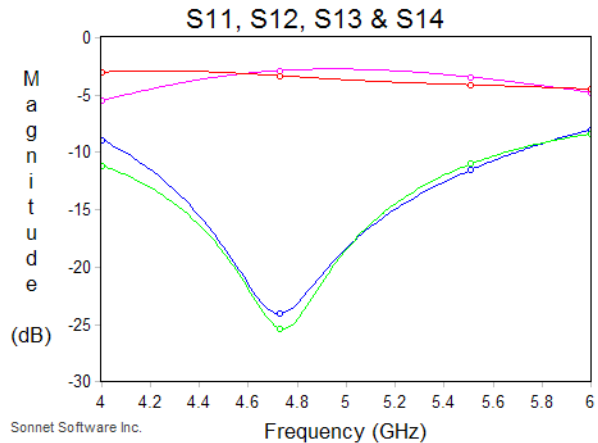
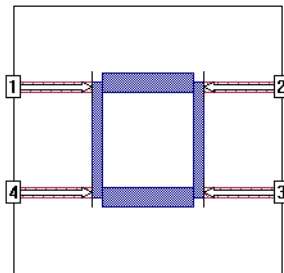
CATEGORY: Multilayer Circuits and Vias

FILE(S): via.son

DESCRIPTION: The circuit shows a microstrip line connected to a via to ground. The top of the via has a pad which is larger than the via itself.

Branchline Coupler

NAME: Branchline



LICENSE REQ'D: Sonnet Lite

CATEGORY: Couplers

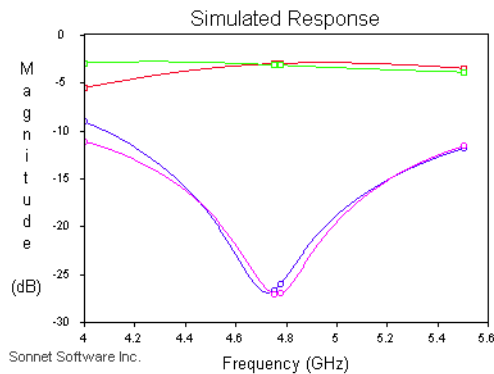
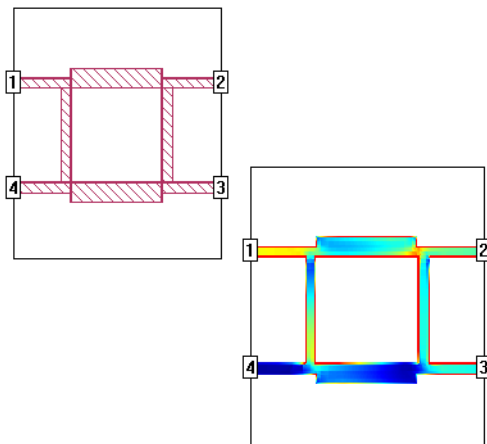
FILE(S): branchline.son

DESCRIPTION: This is an example of a branchline coupler centered at about 4.5 GHz. Port 1 is the input port, port4 is the isolated port, and ports 2 and 3 are the direct and coupled ports. The coupler provides an approximate 3 dB split over a narrow band centered at about 4.5 GHz.

Microstrip Branchline Coupler



NAME: 2Branch



LICENSE REQ'D: Sonnet Lite

CATEGORY: Couplers

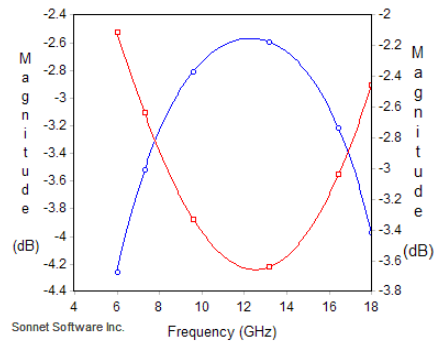
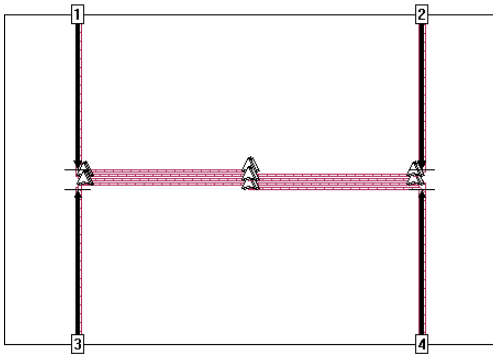
FILE(S): 2branch.son

DESCRIPTION: This example shows a microstrip branchline coupler on alumina ($\epsilon_r = 9.8$) material with a thickness of 25 mils. Analysis also includes a shielded enclosure with metal lid 100 mils above the surface of the alumina substrate.



Lange Coupler

NAME: Lange



LICENSE REQ'D: Sonnet Lite

CATEGORY: Couplers

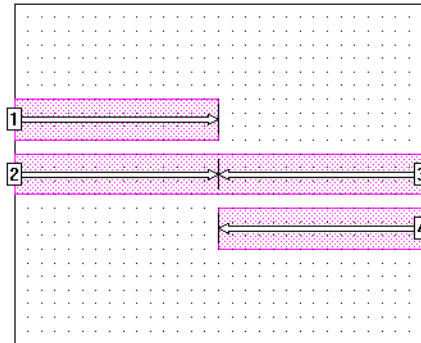
FILE(S): lange.son

DESCRIPTION: This example is a Lange coupler on 15 mil alumina. The coupler is designed to work from 6 - 18 GHz.

Coupled Line Discontinuity



NAME: Coup_end



LICENSE REQ'D: Sonnet Lite

CATEGORY: Filters

FILE(S): coup_end.son

DESCRIPTION: This example shows the discontinuity between two sections in the classic microstrip bandpass filter. You can combine circuit-theory models of your coupled lines with Sonnet S-parameters of each discontinuity to obtain accurate S-parameters for the complete filter.

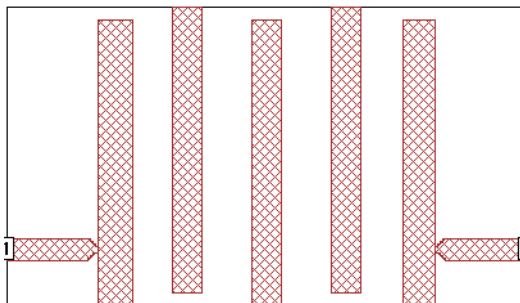
In contrast with an entire filter structure, the coupled line discontinuity results vary slowly with frequency. Thus, an analysis of the discontinuity at three of four frequencies is all that is needed. A circuit theory program may then be used to interpolate and assemble the entire, frequency sensitive, filter.

TIPS It is possible to analyze the entire filter with Sonnet. See [Chapter 14, "Circuit Subdivision"](#) in the **Sonnet User's Guide** for an example of using circuit subdivision to accomplish this task.

Microstrip Interdigital Bandpass Filter



NAME: Filtwall



LICENSE REQ'D: Sonnet Lite

CATEGORY: Filters

FILE(S): filtwall.son

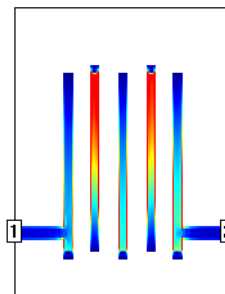
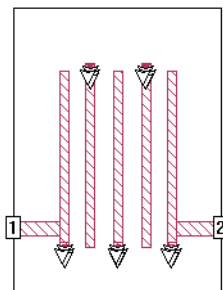
DESCRIPTION: This is an example of a microstrip interdigital bandpass filter with a center frequency of 5.1 GHZ and a bandwidth of 340 MHZ. The filter consists of two impedance transforming sections on the left and right sides and 3 resonator elements in between. Each resonator element is a quarter-wavelength long at the midband frequency and is short-circuited at one end and open-circuited at the other end. Note that the short-circuiting of the resonator elements is achieved in this example by attaching the ends of the elements to the top and bottom box walls. All box walls in Sonnet are perfect grounds.

This circuit is used in the first Sonnet tutorial. Refer to [Chapter 4, "Tutorial: A Quick Tour"](#) in the **Getting Started** manual.

Microstrip Interdigital Bandpass Filter



NAME: Intdigbpfiler



LICENSE REQ'D: Sonnet Lite

CATEGORY: Filters

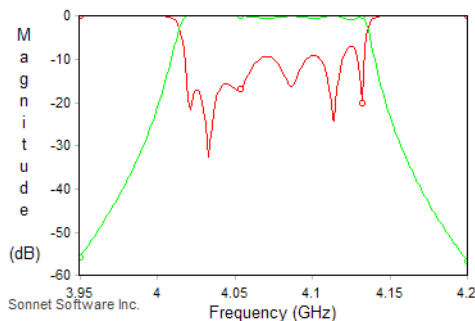
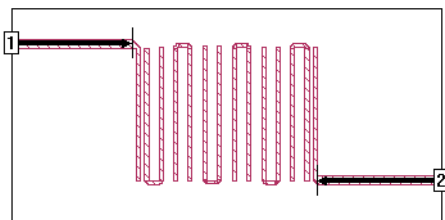
FILE(S): intdigbpfiler.son

DESCRIPTION: This example shows an interdigital bandpass filter on $\epsilon_r = 2.0$ material with a thickness of 40 mils. Resonator conductors are 80 mils wide, and about 1465 mils long. Spacing between outer resonators is 140 mils, and spacing between inner resonators is 160 mils. Vias to ground tap opposite ends of adjacent resonators. Analysis also includes a shielded enclosure with metal lid 400 mils above the surface of the dielectric substrate.



Hairpin Filter

NAME: Hairpin



LICENSE REQ'D: Sonnet Lite

CATEGORY: Filters

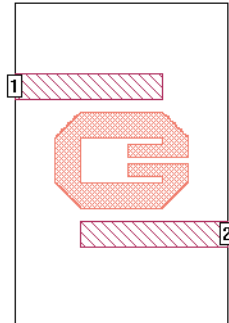
FILE(S): hairpin.son

DESCRIPTION: This is an example of a hairpin filter with a passband of approximately 4.0 to 4.15 GHz. The Speed/Memory slider bar has been set to the middle setting, which reduces the memory requirement significantly. This is also a good example to use to compare non-ABS and ABS results as shown in the plot above, where the non-ABS and ABS curves are nearly identical, although the ABS sweep required analysis at only four frequency points.

TIP: For Sonnet Level2 Basic and above, diagonal fill may be used to improve the diagonal edges of the bends.

Resonator

NAME: Resonator



LICENSE REQ'D: Sonnet Lite

CATEGORY: Filters

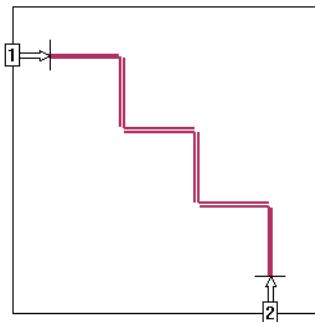
FILE(S): resonator.son

DESCRIPTION: This is an example of a single compact resonator. Metal loss has been included in the resonator, but not in the feed lines, so only the resonator loss is evaluated.

Stripline Coupled Line Filter



NAME: Slcoup



LICENSE REQ'D: Sonnet LitePlus

CATEGORY: Filters

FILE(S): slcoup.son

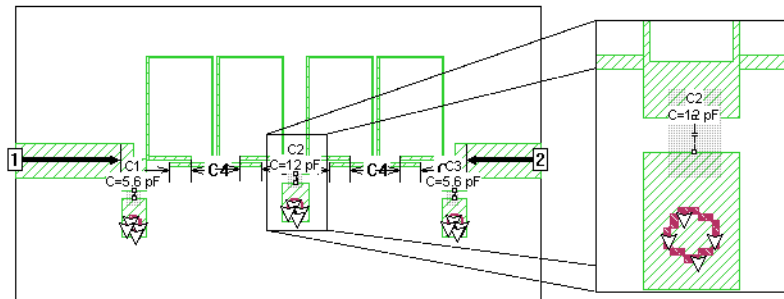
DESCRIPTION: This example is a stripline coupled line filter. The stripline is composed of two 25 mil RT/duroid(R) 6010 dielectric layers. The filter is composed of six coupled line sections.

TIP: The Speed/Memory control has been set to the fastest setting for this example. For a more accurate analysis, set move the control to the left, toward the higher accuracy setting. This requires a Sonnet LitePlus or above. In addition, diagonal fill could be used for the bends (Sonnet Level2 Basic or above required). Since this circuit has a large box (1200 by 1200 cells), using a binary box would make this circuit run faster.

Low Pass Filter with Ideal Element Components



NAME: Component_lpf



LICENSE REQ'D: Sonnet Level2 Basic

CATEGORY: Filters

FILE(S): component_lpf.son

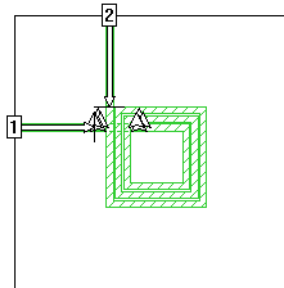
DESCRIPTION: This is a low pass filter consisting of printed-L and ideal component capacitors. The printed-L capacitors are illustrated by the parameters labeled C4. The ideal component capacitors are labeled C1 (5.6 pF), C2 (12.0 pF), and C3 (5.6 pF). This example illustrates the use parameters and ideal components within a Sonnet simulation.

TIP: The Sonnet Professional suite allows you to use the Data File and Ports Only component models in addition to the Ideal Element component model.

Spiral Inductor with Efficient Cell Size



NAME: Spiral



LICENSE REQ'D: Sonnet Lite

CATEGORY: Inductors

FILE(S): spiral.son

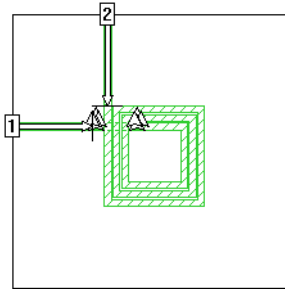
DESCRIPTION: This example circuit is a three turn spiral inductor on 25 mil alumina. The metalization under the alumina provides the ground for the spiral inductor ports. Any capacitance from the spiral to the ground is accurately modeled with Sonnet. This circuit is used in the **Getting Started** manual as a demonstration in determining the optimal cell size for your circuit as well as using the palette of standard geometries and vias. See [Chapter 6. "Determining Cell Size"](#) in the **Getting Started** manual.

The original spiral dimensions can be seen by loading [spi_exact.son](#).

Spiral Inductor with Inefficient Cell Size



NAME: Spi_exact



LICENSE REQ'D: Sonnet Level3 Gold

CATEGORY: Inductors

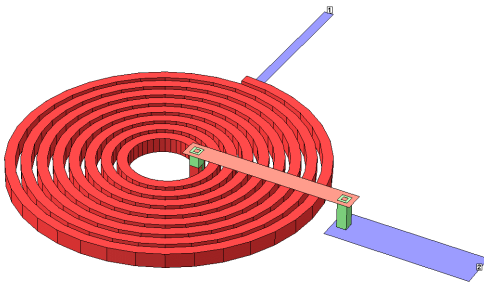
FILE(S): spi_exact.son

DESCRIPTION: This example circuit is a three turn spiral inductor. The spiral has a conductor width of 9.72 mils and spacing is 3.4 mils, with an overall size is 150 mils by 150 mils. See [spiral.son](#) for a more efficient cell size to use for this circuit. The Sonnet tutorial explains the procedure for selecting a more efficient cell size. See [Chapter 6, "Determining Cell Size"](#)

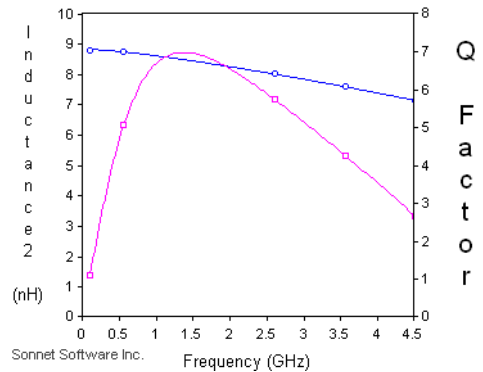
Thick Metal Inductor on Silicon



NAME: Circ_spi



3D view of spiral (View \Rightarrow View 3D)



A plot of inductance versus the Q of the inductor. The response viewer's equation feature was used to create this plot.

LICENSE REQ'D: Sonnet Professional

CATEGORY: Inductors

FILE(S): circ_spi.son

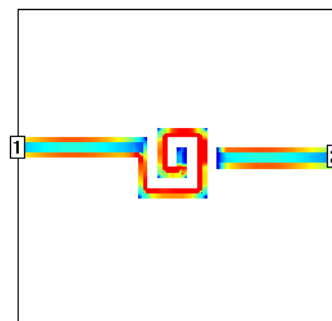
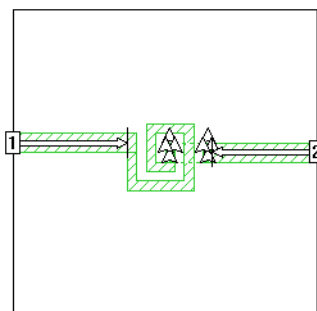
DESCRIPTION: This example is an 8.25 turn spiral inductor on doped silicon. An extra shielding layer is placed above the doped silicon to help isolate the spiral inductor metal from the lossy substrate. Sonnet recommends using a thick metal model for the spiral traces when the thickness of the spiral metal is similar to or greater than the width or spacing of the spiral traces. In this case, the spacing and thickness are both 4 microns, so thick metal is recommended.

The cell fill for the spiral uses conformal mesh to more efficiently model the curved traces. Without conformal mesh, this circuit would require about 5 GB for an accurate simulation. Using conformal mesh reduces the memory requirement to less than 30 MB, while maintaining the accuracy of a 1 micron cell size. For more information on conformal meshing, please see Chapter 12, "Conformal Mesh," in the **Sonnet User's Guide**. There is also an App Note on conformal mesh in online help.

2-Turn Spiral Inductor with Air Bridge



NAME: Indtwoturn



LICENSE REQ'D: Sonnet Lite

CATEGORY: Inductors

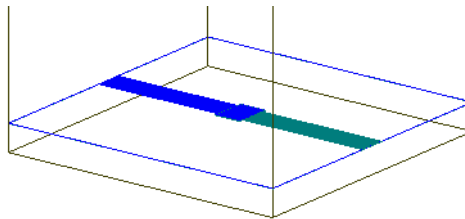
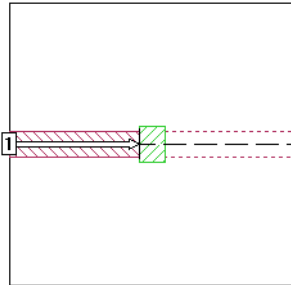
FILE(S): Indtwoturn.son

DESCRIPTION: This example shows a 1.5-turn microstrip series-configuration spiral inductor on undoped silicon ($\epsilon_r = 11.9$). Width and spacing of the inductor are both 10 mils. An air bridge with a height of 2 mils connects the center tap to the output microstrip line.

Shown above on the left is the layout of the circuit feed lines and spiral on the lower level in the project editor, sitting on a 10 mil Si substrate with ground plane below. Upward-pointing triangles indicate vertical vias going up from this layer. Dark arrows indicate reference planes for automatic de-embedding of the port discontinuities and connecting feed lines. The two vias support the metal bridge in air, on the layer above (not pictured).

Simple Capacitor

NAME: Cap



LICENSE REQ'D: Sonnet Lite

CATEGORY: Capacitors

FILE(S): cap.son

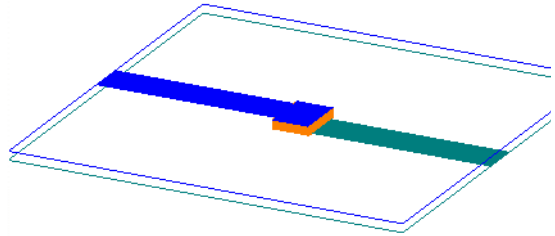
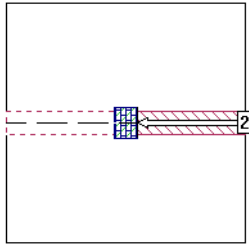
DESCRIPTION: This example is a MIM capacitor which is composed of two parallel plates with a dielectric layer between the plates. A microstrip line is connected to each plate. The dielectric layer is used to represent the thin (0.2 micron) silicon nitride between the capacitor plates.

TIP: This circuit contains a dielectric layer which encompasses the entire circuit. This introduces some error if the real circuit does not have silicon nitride over the entire circuit. However, for thin layers, this error may be undetectable. See also brickcap.son for a capacitor which uses a dielectric brick between the two dielectric plates.

Capacitor using Dielectric Brick



NAME: Brickcap



LICENSE REQ'D: Sonnet Professional

CATEGORY: Capacitors

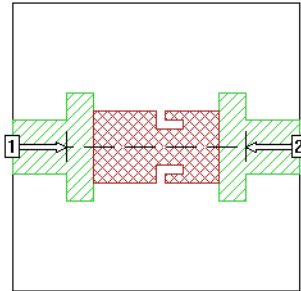
FILE(S): brickcap.son

DESCRIPTION: This example is a MIM capacitor which uses a dielectric brick. The capacitor is composed of two parallel plates with a dielectric brick between the plates. A microstrip line is connected to each plate. The dielectric brick is used to represent the thin (0.2 micron) silicon nitride between the capacitor plates.

TIP: Dielectric bricks can use a large amount of memory if used extensively. This capacitor could also be created by deleting the dielectric brick and setting the dielectric constant of the entire 0.2 micron layer to be silicon nitride ($\epsilon_r = 6.8$). This introduces some error if the real circuit does not have silicon nitride over the entire circuit. However, for thin layers, this error may be undetectable. See also [cap.son](#) for a capacitor which does not use dielectric bricks.

Notched Resistor

NAME: Notch_resistor



LICENSE REQ'D: Sonnet Lite

CATEGORY: Resistors

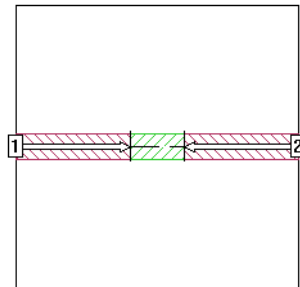
FILE(S): notch_resistor.son

DESCRIPTION: This example is of a notched resistor (tuned by a laser, perhaps). The circuit uses metalization loss in both the resistor material and in the copper leads.

TIPS: The resistor is approximately 10 ohms. This is easiest seen by viewing the real part of Y12. This shows a series admittance of about 0.1 Siemens.

Thin Film Resistor

NAME: Res400



LICENSE REQ'D: Sonnet Lite

CATEGORY: Resistors

FILE(S): res400.son

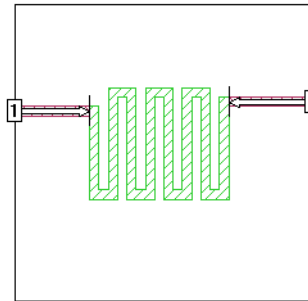
DESCRIPTION: This example illustrates the modeling of a thin film resistor in Sonnet. The circuit contains two microstrip transmission lines with a thin film resistor in the middle. A characteristic of thin film resistors is that their skin depth is usually much greater than their thickness. Thus, they can be modeled by using the Resistor metal type with an appropriate value for Rdc (frequency independent loss). In this example, the resistor metallization is $120\text{ }\mu\text{m}$ long and $60\text{ }\mu\text{m}$ wide. Therefore, an Rdc value of 200 ohms/sq is required to obtain a total resistance of 400 ohms.

TIPS: On conductors which are nearly lossless, such as the two transmission lines in this example, current travels primarily along the outer edges of the conductor. Thus, enabling edge mesh for these conductors can reduce simulation time with little loss in accuracy. However, on the thin film resistor the current will tend to be much more distributed. Therefore, we recommend that you do not enable edge mesh for resistor polygons as it may result in some loss of accuracy.

Resistor using Meandering Lossy Metal



NAME: Res500



LICENSE REQ'D: Sonnet Lite

CATEGORY: Resistors

FILE(S): res500.son

DESCRIPTION: This circuit represents a resistor composed of a meandering length of lossy transmission line. Since Sonnet is a full-wave analysis, all electromagnetic effects are accurately modeled. This can be seen by viewing the SPICE lumped model of the circuit, which includes some shunt parasitic capacitances.

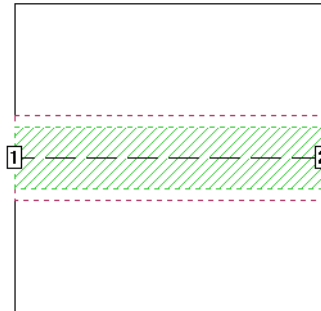
For a straight resistor you can count the number of squares and multiply by the sheet resistivity (in ohms per square) to determine the total resistance at low frequency. However, when the line is bent, as in this example, the number of squares is no longer known (because of current crowding at the bend). Sonnet can be used to quickly determine the equivalent resistance for this case.

TIP: This example uses a metal which has 5 ohms per square resistivity at all frequencies of interest. This is valid if the conductor is thin with respect to a skin depth at the highest frequency of analysis. See "[Metalization Loss](#)" for more information about accurately modeling the loss of metal conductors.

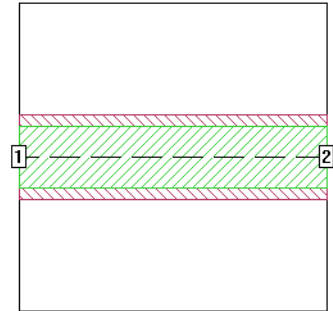
Thick Metal Through Line



NAME: Thkthru



Level 0



Level 1

LICENSE REQ'D: Sonnet Level2 Basic

CATEGORY: Techniques

FILE(S): thkthru.son

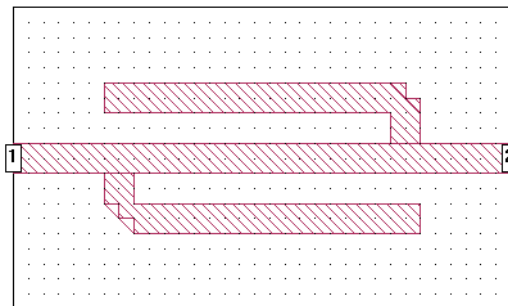
DESCRIPTION: The example shows a thick metal through line. This is modeled in Sonnet using the thick metal type. A 3 mil thick conductor is placed on top of a wider zero-thickness conductor (the cross-section is actually trapezoidal). Notice that the thick metal appears on both level 1 where it was drawn and on level 0 because the dielectric layer was also 3 mils thick. Sonnet recommends attaching ports to all polygons even though, as in this case, ports are automatically propagated to polygons which overlap on the same metal level.

TIP: By default, Sonnet models metal as infinitely thin using a thickness-dependant surface impedance to model the loss. This is the most efficient method to model thickness. However, the above technique may be used if the thickness of the metal becomes large (for instance, 10% of the substrate thickness). See also the [thkstep](#) example. For a detailed discussion of thick metal, see [Chapter 17, "Thick Metal"](#) in the **Sonnet User's Guide**.

Double Folded Stub Tuner



NAME: Dstub



LICENSE REQ'D: Sonnet Lite

CATEGORY: Tutorial

FILE(s): dstub.son

DESCRIPTION: This example circuit is used in the Sonnet tutorial to teach the basics of creating and analyzing a circuit. It is a simple microstrip double stub tuner. The circuit is composed of a microstrip through line with two quarter-wave stubs placed approximately a quarter wave apart. The stubs have been folded to conserve space. Since Sonnet is a full-wave analysis, the coupling between the line and the stubs, and the effects of the bends are included in the analysis.

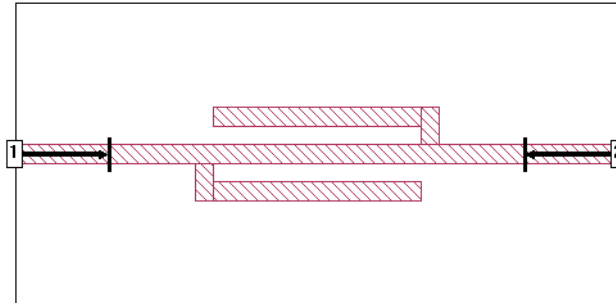
Note that the tutorial, which creates the project file from a new project, uses a linear frequency sweep for the analysis, but that this example file uses an adaptive sweep (ABS) for the analysis.

TIP: The analysis engine always analyzes the fill pattern of the polygon, not the actual polygon. Notice the fill pattern around the bends shows a “staircase” pattern. This means that the analysis engine will analyze a staircase approximation to the actual diagonal edge. To improve the accuracy of the analysis, use diagonal fill to better represent the diagonal edge. Diagonal fill is only necessary when the cell size is large with respect to the bend dimensions. As the cell size becomes smaller, the default staircase fill gives the same results as diagonal fill.

Double Folded Stub Tuner w/Parameters



NAME: Par_dstub



LICENSE REQ'D: Sonnet Lite

CATEGORY: Tutorial

FILE(S): par_dstub.son

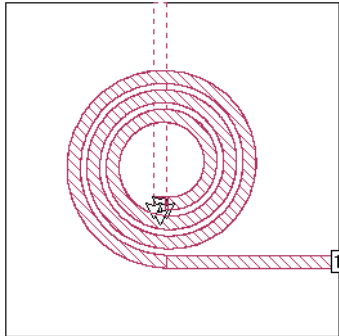
DESCRIPTION: This example circuit is used in the Sonnet tutorial to teach the basics of parameterization and optimization. It is a simple microstrip double stub tuner. The circuit is composed of a microstrip through line with two quarter-wave stubs placed approximately a quarter wave apart. The stubs have been folded to conserve space. Since Sonnet is a full-wave analysis, the coupling between the line and the stubs, and the effects of the bends are included in the analysis.

This circuit is used in the parameterization and optimization tutorial and is based on the circuit, [Dstub](#), used in [Chapter 5, "Creating a Circuit: Dstub"](#) in the **Getting Started** manual. You may also refer to [Chapter 11, "Parameter Sweep and Optimization Tutorial."](#) in the **Sonnet User's Guide**.

Circular Spiral with Conformal Meshing



NAME: Cm_spiral



LICENSE REQ'D: Sonnet Professional

CATEGORY: Conformal Mesh

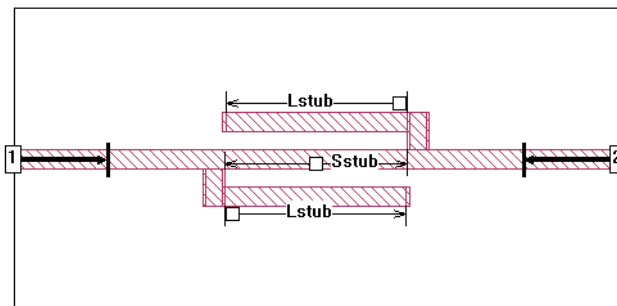
FILE(S): cm_spiral.son

DESCRIPTION: This example is used in the Conformal Mesh App Note available in online help. Note that this example contains several problems which are corrected during the course of the tutorial. We do not recommend running an analysis on this circuit without performing the modifications detailed in the tutorial.

Optimized Double Folded Stub Tuner



NAME: Par_ds_opt



LICENSE REQ'D: Sonnet LitePlus

CATEGORY: Tuning

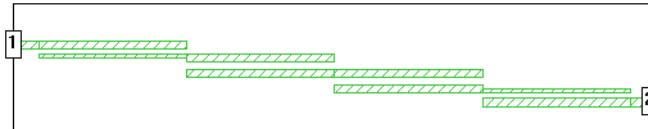
FILE(S): par_ds_opt.son

DESCRIPTION: This example is the same as [par_dstub](#) except that the parameters and analysis setup added in [Chapter 11, "Parameter Sweep and Optimization Tutorial,"](#) in the **Sonnet User's Guide** are included in this example.

Bandpass Filter for Circuit Subdivision



NAME: Subdivide



LICENSE REQ'D: Sonnet Level2 Basic

CATEGORY: Tutorial

FILE(s): subdivide.son

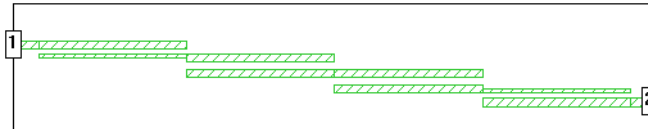
DESCRIPTION: This is an example of a bandpass filter used to demonstrate the circuit subdivision technique using Sonnet software. Note that the project is not ready to analyze; changes made in the tutorial mentioned below complete the specification of the project.

This circuit is used in the circuit subdivision tutorial. Refer to [Chapter 15. "Circuit Subdivision Tutorial,"](#) in the **Sonnet User's Guide**.

Source Analysis for Circuit Subdivision



NAME: Sub_whole



LICENSE REQ'D: Sonnet Level2 Basic

CATEGORY: Tutorial

FILE(s): sub_whole.son

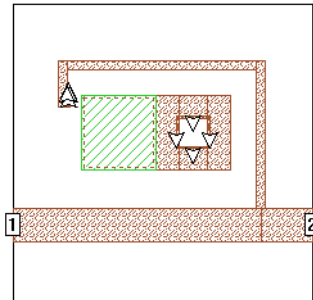
DESCRIPTION: This is an example of a bandpass filter used to demonstrate the circuit subdivision technique using Sonnet software. This project contains the analysis of the source circuit to compare with the results of the netlist analysis done in the circuit subdivision tutorial.

This circuit is used in the circuit subdivision tutorial. Refer to [Chapter 14. "Circuit Subdivision"](#) in the **Sonnet User's Guide**

Short Circuited Shunt Connected Stub



NAME: Raystub



LICENSE REQ'D: Sonnet Lite

CATEGORY: Multilayer Circuits and Vias

FILE(S): raystub.son

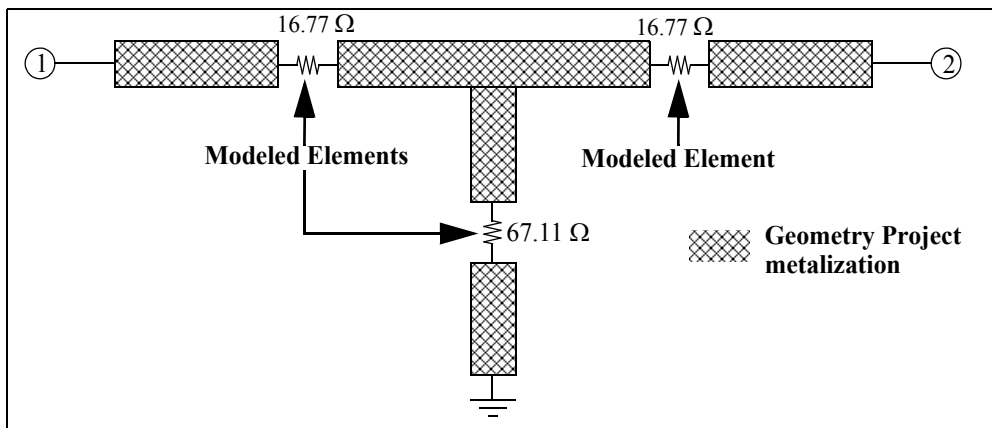
DESCRIPTION: This is an example of a short circuited shunt connected stub. The stub wraps around, connects to a MIM capacitor, and then is connected to ground with a via. Notice the construction of the capacitor. The top plate is located on metallization level 0 and the bottom plate on level 1. The dielectric layer between the two plates is 0.20 μm thick and has a relative dielectric constant of 6.8. Notice that this dielectric layer covers the entire area of the box, not just the area between the capacitor plates. Because of this, there is a slight error in the fringing capacitance. Typically, this error is very small compared to the total capacitance, and thus, this is the preferred method for modeling parallel plate capacitors in Sonnet. If the fringing capacitance is important, an alternative method is to set relative epsilon for the dielectric layer equal to 1.0 and then insert a dielectric brick between the capacitor plates. However, dielectric bricks can be computationally intensive, so this method should only be used when accurately modeling the fringing capacitance is important.



COPY EXAMPLE

Attenuator

NAME: Att



LICENSE REQ'D: Sonnet Professional for att_lumped.son. Sonnet Level3 Gold for att_lgeo.son. Sonnet Level2 Basic for att_lumped2.son. Sonnet LitePlus for att_lgeo2.son. Sonnet Lite for all other files.

CATEGORY: Circuit Theory, Resistors

FILE(S): att_combine.son, att_cascade.son, att_lgeo.son, att_lumped.son, att_lgeo2.son, att_lumped2.son, att_res16.son, att_res16.s2p, att_res67.son

DESCRIPTION: Example files used in [Chapter 13, "Netlist Project Analysis"](#) in the **Sonnet User's Guide** to explain how to use the netlist features in Sonnet. The example is a T attenuator, composed of two 16 ohm resistors and one 67 ohm resistor. Each of the files is described below:

att_res16.son: Approximately 16 ohm resistor with feed lines.

att_res67.son: Geometry of a 67 ohm resistor with feed lines.

[Continue](#)



COPY EXAMPLE

Attenuator (Cont'd)

att_res16.s2p: S-parameter results file of att_res16.son

att_cascade.son: Netlist project which connects two S-parameter files together.

att_combine.son: Netlist project which creates a complete attenuator by connecting S-parameter files (att_res16.s2p) to a geometry project (att_res67.son).

att_lgeo.son: Geometry project using autogrounded ports to allow adding of lumped resistors.

att_lumped.son: Netlist project which connects att_lgeo.son to lumped resistors.

att_lgeo2.son: Geometry project using internal ungrounded ports to allow adding of lumped resistors.

att_lumped2.son: Netlist project which connects att_lgeo2.son to lumped resistors.

NOTE:

This example may also be analysed by using components to model the resistors in the attenuator. For more information on Components, please see [Chapter 6, “Components,”](#) in the Sonnet User’s Guide,

[Back](#)

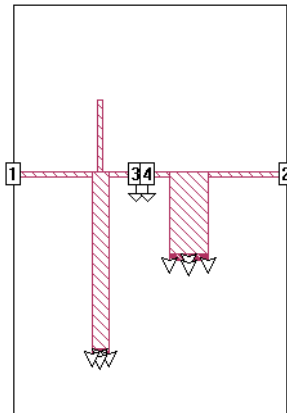
[Next Example](#)

Amplifier Using Measured Device Parameters



COPY EXAMPLE

NAME: Amp



LICENSE REQ'D: Sonnet Lite

CATEGORY: Circuit Theory

FILE(S): ampgeo.son, ampnet.son, amp_dev.s2p

DESCRIPTION: This example is a simple amplifier composed of the main netlist project, ampnet.son, and a geometry project, ampgeo.son. The geometry project uses autogrounded ports to allow the device measured S-parameters (amp_dev.s2p) to be inserted in the middle of the circuit. This is accomplished in the netlist project, ampnet.son.

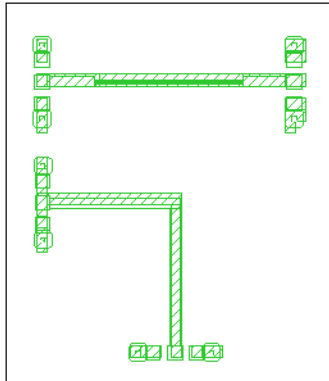
TIP: The netlist project is analyzed at 2 to 20 GHz in steps of 0.1 GHz, but the geometry project is analyzed using ABS over the 2 -20 GHz frequency band to save EM analysis time. The data from the geometry project is then connected to the device, giving final S-parameters with 0.1 GHz steps.

GDSII & DXF Translator Project - Multiple Files



COPY EXAMPLE

NAME: Dcblock



CATEGORY: GDSII & DXF Translator Example Files

FILE(S): dcblock.dxf, dcblock.gds, dcblock.lay

DESCRIPTION: Example files used in the GDSII and DXF translator examples.

Ebridge - ADS Project Folder



COPY EXAMPLE

NAME: Ebads_prj

CATEGORY: Ebridge to Agilent ADS Example Files

FILES: Multiple Files

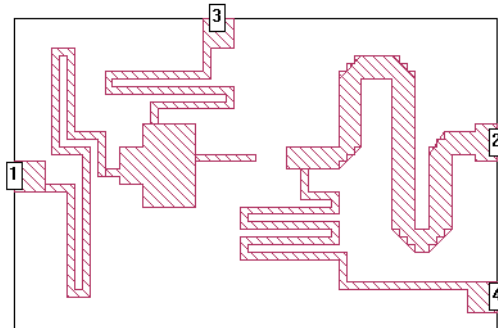
DESCRIPTION: Example files used in the Agilent ADS examples.

Package Resonance Files



COPY EXAMPLE

NAME: Package_resonances



LICENSE REQ'D: Sonnet Lite

CATEGORY: Techniques

FILE(s): open_res.son, open_nores.son, and package.son

DESCRIPTION: This set of circuits is discussed in [Chapter 22, "Package Resonances"](#) of the **Sonnet User's Guide**. The file package.son has a box resonance at about 31.76 GHz. The circuit open_res.son has a box resonance at about 24.3 GHz. The file open_nores.son uses a free space top cover to remove the box resonance.

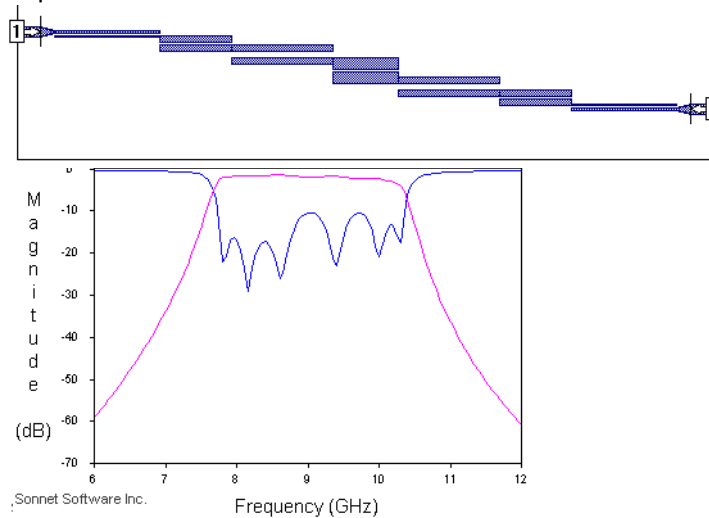
Edge-Coupled Microstrip Bandpass Filter



COPY EXAMPLE

NAME:

Bpfilter



LICENSE REQ'D: Sonnet Level2 Silver for bpf_whole.son. Sonnet LitePlus for halfgeo.son. Sonnet Lite for all other files.

CATEGORY: Filters, Circuit Theory

FILE(s): bpf_main_net.son, bpf_whole.son, halfgeo.son, halfnet.son, halfnet_s1.son, halfnet_s2.son, halfnet_s3.son, halfnet_s4.son, bpf_meas_s21.s2p

DESCRIPTION: This example is a seven-section microstrip bandpass filter which uses circuit subdivision to reduce the analysis time and memory required. It is included as an example here to show how to take advantage of the symmetry of the filter. Since the left half is identical to the right half, you can use circuit subdivision on the left half of the circuit, and reuse the

[Cont'd](#)

Edge-Coupled Microstrip Bandpass Filter



COPY EXAMPLE

subdivided pieces for the right half. This saves an additional factor of two in analysis time. For Level2 Silver users and above, this process is simplified by using the circuit subdivision feature of Sonnet:

To analyze this filter, you would do the following:

1. Create a geometry project (halfgeo) which represents the left half of the whole filter (bpf_whole).
2. Add subdividers to halfgeo and subdivide the circuit (*Tools* \Rightarrow *Subdivide*). This creates a netlist project (halfnet), and four sub-projects (halfnet_s1, halfnet_s2, etc.).
3. Edit the netlist project, halfnet, and add a new network which represents the two halves cascaded together. See bpf_main_net, where the new network, “whole” represents the two half-networks connected together.
4. Run the new project, bpf_main_net to obtain the results for the entire filter.

NOTE:

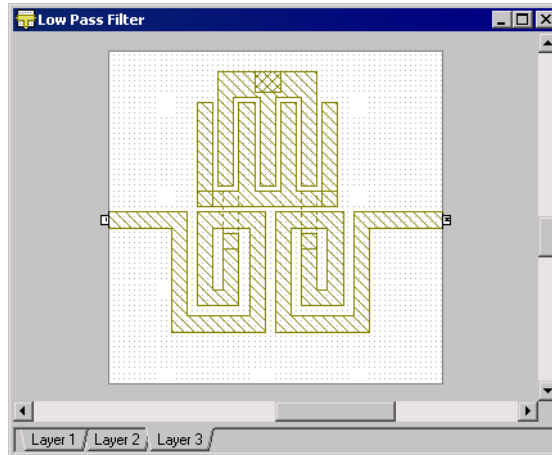
The analysis results for the entire filter without using circuit subdivision have been included in this example set so you can compare the two methods. See bpf_whole. See also bpf_meas_s21.s2p for S21 measurements of this filter. Design and measurements of this filter were provided by Kaman Sciences, Inc.

[Back](#)

Microwave Office Project



NAME: Lowpass



LICENSE REQ'D: Sonnet Lite for EM structure **Low Pass Filter**

CATEGORY: Microwave Office Interface

FILE(S): lowpass.emp

DESCRIPTION: This is an example of an AWR Microwave Office project which references a Sonnet geometry project. The overall circuit is a low-pass filter consisting of three identical stages cascaded together. The Sonnet project, or EM structure, is one of those stages. When the overall circuit is analyzed, Sonnet is called from Microwave Office just once, to analyze one of the stages. Microwave Office then uses the Sonnet results three times to obtain the response for the overall low-pass filter.

Stripline Benchmark Excel Spreadsheet



OPEN EXCEL
SPREADSHEET
(WINDOWS ONLY)

NAME: Csl_xls

LICENSE REQ'D: Not Applicable

CATEGORY: Microsoft Excel

FILE(S): csl.xls

DESCRIPTION: This is an Excel spreadsheet which contains the long-known exact solution for lossless infinitely thin stripline. This is provided for the application note, Coupled Stripline Benchmark, available in online help. Also, you may refer to [Chapter 23, "Accuracy Benchmarking"](#) in the **Sonnet User's Guide** for a discussion of the single stripline standard benchmark. You may access the manuals in PDF format through the Sonnet task bar.

Gerber Translator Project - Multiple Files



COPY EXAMPLE

NAME: Gerber_ex



CATEGORY: Gerber Translator Example Files

FILE(s): m1.gbr, m2.gbr, Drill_thru.gbr, DRILL_THRU.drl, drill_table_report.txt,
Gerber_Tutorial_Full_Board.njb

Description:

Example files used in the Gerber translator tutorial. Please see [Chapter 6, "Gerber Translator Tutorial,"](#) on page 245 of the Translators manual.